# An active subspace method for accelerating convergence in Delaunay-based optimization via dimension reduction

Muhan Zhao ⋆ Shahrouz Ryan Alimo⋆⋆ Thomas R. Bewley†

***Abstract*— Delaunay-based derivative-free optimization, Δ-DOGS, is an efficient and provably-convergent global optimization method for the problems which has computationally-expensive objection function and the analytical expression for the objective function is not available. Δ-DOGS is a novel optimization scheme in the family of response surface methods (RSMs); however, it suffers from the curse of dimensionality since the computational cost increases dramatically as the number of design parameters increases. As a result, the number of design parameters in Δ-DOGS algorithm is relatively low ($n \lesssim 10$). To avoid such problems, this paper proposes a combination of derivative-free optimization, seeking the global minimizer of an expensive and nonconvex objective function $f(x)$ and active subspace method, detecting the directions of the most variability using evaluations of the gradient. The contribution of other directions to the objective function is bounded by a sufficiently small constant. This new algorithm iteratively applied Delaunay-based derivative-free optimization to seek the minimizer on the $d$-dimensional active subspace that has most function variation. Inverse mapping is needed to project data from active subspace to full-model for evaluating function values. This task is overcome by solving an inequality constrained problem that curves the response surface of the objective function. The test results show that this strategy is effective on a handful of optimization problems.**

## I. Introduction

In this paper, we consider a nonconvex optimization problem as follows:

$$\text{minimize } f(x) \text{ with } \; x \in B = \{x | a \leq x \leq b\}, \qquad (1)$$

where $a$ and $b$ are two vectors in $\mathbb{R}^n$ such that $a < b$, and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is expensive-to-compute. We seek a point $x \in B$ such that the function value of $f(x)$ is less than or equal to $f_0$. Solving an optimization problem of the form (1) is difficult and, for general functions, convergence can only be guaranteed if the function evaluation set becomes dense over the entire search domain, $B$, in the limit of an infinite number of function evaluations [1], [2].

In this paper, for the purpose of derivation of our algorithm, we thus focus our attention on problems in which $n$-dimensional objective function $f(x)$ varies most along a few $d$ directions, while the other $n-d$ directions only have a small contribution bounded by a sufficiently small constant $\gamma$.

⋆ Dept of MAE, UCSD, muz021@ucsd.edu

⋆⋆ Jet Propulsion Laboratory, California Institute of Technology, sralimo@jpl.nasa.gov

‡ Professor of Dept of MAE, UCSD, bewley@ucsd.edu

The nonconvex objective function gained lots of interests in engineering problems recently, such as the optimization of the compliance properties of fabrics for improved flow/structure interactions (reduced drag, reduced noise, etc.) and hydrofoil design optimization. The hydrofoil optimization problem described in [3] represents a typical challenge problem for this effort, as its objective function $f(x)$, which characterizes the lift/drag ratio of the foil, is much more strongly dependent on some of the design parameters than the others. This behavior is common and challengeable in shape optimization since there is no closed form of the objective function, the first-order and second-order derivatives. The heavy cost of each function evaluation computation restricts the number of function evaluations. Therefore, the ultimate goal is to determine the global minimum through as few function evaluations as possible.

For many dimension reduction problems, sensitivity analysis [4], [5] is a well-known method by ranking the input parameters due to the measure of their contribution to the objective function. However, some functions may have the most variable directions that are not aligned with the coordinate found by sensitivity analysis. And the computation cost needed to study the importance of parameters may exceed the available sources.

Principle component analysis [6] is another popular method for dimension reduction by creating new artificial coordinates that are linear combination of the observed variables. PCA could keep as much variation as possible instead of identifying the direction that has most variation of the objective function, which possibly passes the global minimum region. However, derivative optimization iteratively explores the basin of global minimum which is not compatible with the variation of data points.

Locally linear embedding [7] identifies the low-dimensional subspace when the high-dimensional data lie on a manifold that embedded in high dimensional space. It first determines the weights of linearly approximating the data in the neighbors of available data points. And finally finds a low-dimensional coordinates that best reconstruct those weights. However, the subregion of original parameter space is explored by the function values, which indicates that it is required to design a strategy to project the data from low-dimensional embedding back to original parameter space. In Section III-B we propose a retransformation strategy by solving a constrained minimization problem.

Under appropriate assumptions, it is guaranteed that derivative-free methods could converge to a global optimum,

but in general they are computationally inefficient since many more function evaluations are required compared with derivative-based methods at local refinement. Response surface methods (RSMs) are the most efficient globally-convergent derivative-free optimization methods available today. RSMs iteratively minimize a search function using an interpolant of existing data points, known as the "surrogate", and a model of the "uncertainty" of this surrogate which goes to zero at the function evaluations themselves. Efficient global optimization (EGO) [8], optimization by radial basis function interpolation in trust-regions (ORBIT) [9], the Surrogate-Management-Framework (SMF) [10], and Delaunay-based derivative-free optimization via global surrogates (Δ-DOGS) [11], [12], are modern examples of RSMs.

The derivative-free scheme upon which the present work is based on is Δ-DOGS, which is a generalizable family of computationally-efficient RSMs developed by our group to optimize low-dimensional and black-box problems of which the objective function is both nonconvex and computationally expensive. There are already a handful schemes in this family, including schemes designed specifically for simple bound constraints [13], linear constraints [11], [14], and nonconvex constraints [12].

This paper combines the dimensionality reduction scheme together with Δ-DOGS algorithm to minimize the high-dimensional objective function that has most variation along at most $d$-directions. We first apply gradient sampling to obtain the active subspace. Δ-DOGS optimization scheme is applied on the active subspace to identify a low-dimensional minimizer that is potentially close to the global minimum after projection. Then a new inverse mapping scheme is proposed to transform the minimizer back to original parameter space by solving an inequality constrained minimization problem. This new algorithm is global convergence-provable and shows competitive performance in derivative-free global optimization methods.

The paper is structured as follows: Section II briefly reviews the essential ideas of [13], [14], which accelerates a Δ-DOGS search by coordinating it with a Cartesian grid over parameter space that is successively refined as convergence is approached. Section III explains the new optimization scheme, which combines active subspace method with derivative-free optimization scheme. Section IV analyzes the global convergence property of the new algorithm under appropriate assumptions. In Section V, the new algorithm is applied to synthetic optimization problems to illustrate its competitive performance. Conclusions are presented in Section VI.

## II. A BRIEF REVIEW OF Δ-DOGS

In this section we briefly review the essential ideas of Δ-DOGS [13], [14]. Note that this paper focuses on variants of these algorithms that leveraged by active subspace to identify the directions in the parameter space that has most variability; other variants of these algorithms, such as implementing Cartesian grids to accelerate the convergence rate discussed in [13], [14], and multivariate adaptive polyharmonic splines discussed in [15]. Those modification of Delaunay-based optimization could also be incorporated into this framework.

Δ-DOGS algorithm successively determines the location where has the highest probability to achieve the function value equal or less than the given target $f_0$. This approach is realized by minimizing the synthetic and cheap-to-establish surrogate model $s_c(x)$, constructed by Natural Polyharmonic Spline [16] $p(x)$ and the uncertainty function $e(x)$. The approach is akin to the expected improvement [17] and Bayesian optimization algorithms [18].

*Definition 1:* Take $S$ as a set of $N$ points $\{x_i\}_{i=1}^N$ over the feasible domain $L$. Construct the Delaunay triangulation $\Delta$ over the set $S$. The uncertainty function $e(x)$ on each Delaunay simplex $\Delta_i$ is defined as

$$e(x) = R_i^2 - \|x - c_i\|^2, \ \forall \ x \in \Delta_i. \tag{2}$$

Here $R_i$ and $c_i$ are the circumradius and circumcenter of the hypercircumcircle of Delaunay simplex $\Delta_i$.

The uncertainty function approaches maximum within each Delaunay simplex as far from the available data points. In paper[11] several key properties of $e(x)$ has been discussed, including: (a) $e(x)$ is Lipschitz continuous and twice-differentiable within each Delaunay simplex; (b) $e(x) \geq 0 \ \forall x \in B$ and $e(x_i) = 0 \ \forall i \in \{1, \ldots, N\}$.

*Definition 2:* Consider a set of N data points $S = \{x_i\}_{i=1}^N$ over the feasible domain $B$. Suppose the uncertainty function $e(x)$ is established within each Delaunay simplex as defined by equation (2). The continuous search function $s_c(x)$ is defined as follows:

$$s_c(x) = \begin{cases} \frac{p(x) - f_0}{e(x)} & \text{if} \quad p(x) \geq f_0, \\ p(x) - f_0 & \text{otherwise}, \end{cases} \tag{3}$$

where $p(x)$ is some smooth interpolating function such that $p(x_i) = f(x_i), \forall i \in \{1, \ldots, N\}$.

The interpolation $p(x)$, truth function $f(x)$ and continuous search function $s_c(x)$ are illustrated in Fig. 1.

*Definition 3:* The Cartesian grid of level $L$ for the feasible domain $B = \{x | a \leq x \leq b\}$, denoted $B_L$, is defined such that

$$B_L = \left\{ x | x_L = a_L + \frac{1}{N}(b_L - a_L) \cdot z_L, \quad z_L \in \{0, 1, \ldots, 2^L\} \right\}$$

The point $x \in B$ would be quantized onto the grid $B_L$ as $x_q$ which is nearest to $x$ from the grid points. The quantizer $x_q$ is not necessarily unique. Therefore, the maximum quantization error at $\forall x \in B$ to the mesh grid $B_L$ is defined as

$$\delta_L(x) = \max_{x_q \in B_L} |x - x_q| \tag{4}$$

*Remark 1:* At each iteration, Algorithm 1 either evaluates a new feasible data point, or refines the mesh by incrementing $L = L + 1$.

There are two possible termination scenarios for Algorithm 1: either the target value $f_0$ is achievable and Δ-DOGS identifies a point $x$ with function value $f(x) \leq f_0$, or

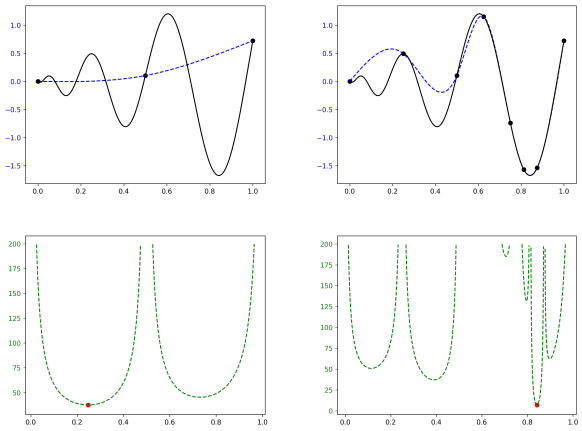

Fig. 1: The essential elements of Δ-DOGS algorithm in different iterations for 1D Schwefel function(30). The upper figures contain: The solid black line indicates the truth function $f(x)$, the blue dotted line indicates interpolant function $p(x)$; The lower figures contain: The green dotted line represents the continuous search function $s_c(x)$, as defined in equation (3). The red circles are the minimizer of $s_c(x)$ as known as the next data point to evaluate.

**Algorithm 1** Strawman of Δ-DOGS, designed for minimizing $f(x) \in B$ leveraging the target value $f_0$.

0. Initialize $k = 0$, $\ell$, and the initial set of data points $S_0$, and calculate $f(x_i)$ for all $x_i \in S_0$.
1. Calculate or update an robust interpolation $p_k(x)$ over the feasible domain $B$.
2. Calculate or update the Delaunay triangulation $\Delta_k$ over the data points $S_k$. Construct the uncertainty function $e_k(x)$ for the points in $S_k$.
3. Find the minimizer $x_k$ of continuous search function (3) in $B$.
4. Determine $y_k$ as the quantization of $x_k$ on $B_{L_k}$.
5. If $y_k \notin S_k$, $S_{k+1} = S_k \cup y_k$, and calculate $f(y_k)$; otherwise, refine the mesh by incrementing $L_k = L_k + 1$. Increase $k = k+1$.
6. Repeat steps 1-5 until a point $x$ is found with $f(x) \leq f_0$.

Algorithm 1 conducts infinite number of mesh refinement iterations to get data points dense in the entire parameter space. In the latter case, it is proved in [13] that the global minimizer is determined as the mesh grid gets successively refined. The quantization error $\delta_L$ would converges to zero since there are only finite number of data points on each mesh grid.

The Algorithm 1 presents a strawman form of the accelerating Δ-DOGS with implementation of Cartesian grid from the above concepts. Further details are presented in [13], [14].

Despite that Δ-DOGS is a family of computationally efficient schemes for globally exploring a large range of non-convex problems through surrogate minimization, it suffers from "the curse of dimensionality" as all derivative-free optimization schemes, and scales poorly with the dimension of the problem. The dimension reduction algorithm proposed below mitigates this issue.

## III. Dimension Reduction Based on Active Subspace Method

### A. Active Subspace method

In this section we briefly discuss the theory of active subspace method [19]. Note that principle component analysis implements the proper orthogonal decomposition on the covariance matrix of original data. In this way PCA obtains the most variate component that is a linear combination of original coordinate system. Similar to principle component analysis, active subspace method performs the proper orthogonal decomposition but on the covariance matrix of the gradient of the objective function.

Consider the scalar function $f$ on the $n$-dimensional column vector $x$, whose variability is concentrated in $d$ directions. The gradient $\nabla f(x)$ is also reshaped as a column vector. Suppose the evaluated data points set is denoted as $S = (x_1, x_2, ..., x_N)$,

$$f_i = f(x_i),\ \nabla f(x_i) \in \mathbb{R}^n,\ x_i \in \mathscr{X} = [0,1]^n \tag{5}$$

The task is to identify the active directions that effectively represents the most variability of $f$. Let $\rho = \rho(x)$ be the uniform probability density function. The covariance matrix is estimated by randomly sampling the gradient in the parameter space with Monte Carlo method. The estimated covariance matrix is

$$C \approx \hat{C} = \frac{1}{M}\sum_{i=1}^{M} \nabla f(x_i) \nabla f(x_i)^T \tag{6}$$

The active directions of the parameter space is determined by performing the spectral decomposition on covariance matrix which is symmetric and positive semidefinite.

$$C = W \Lambda W^T \tag{7}$$

where $W \in \mathbb{R}^{n\times n}$ and $\Lambda$ is a diagonal matrix of descending eigenvalues. The first $d$ eigenvectors from $W$ are selected to form the *active subspace*. Notice that their corresponding eigenvalues are relatively large, which means there are more variability along the direction denoted by those eigenvectors. Since we consider the objective function only varies along $d$ directions, the relatively small eigenvalues will be ignored.

From the assumption that the most variability of objective function is along $d$ directions in the parameter space, we determine the gap in eigenvalues matrix $\Lambda$ and partition the eigenvectors matrix $W$ and $\Lambda$. Up till now we have all the essential elements to construct the active subspace. The framework is described in Algorithm 2.

$$W = \begin{bmatrix} \underbrace{W_1}_{d \text{ columns}} & \underbrace{W_2}_{n-d \text{ columns}} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \tag{8}$$

Once the spectral decomposition of matrix $C$ is acquired, the original parameter $x$ can be viewed as $x = W_1^T y + W_2^T z$,

**Algorithm 2** Strawman of Active Subspace Method

0. Draw $M = \alpha k \log(n)$ samples $x_j$ independently according to the density function $\rho$.
1. For each $x_j$, compute the gradient

$$\nabla f(x_i) = \left[ \frac{\partial f}{\partial x_{i1}} , \frac{\partial f}{\partial x_{i2}} , \ \ldots \ , \ \frac{\partial f}{\partial x_{in}} \right]^T , \quad i \in \{1, ..., M\}$$

.
2. Approximate by Monte Carlo sampling

$$C \approx \hat{C} = \frac{1}{M} \sum_{i=1}^{M} \nabla f(x_i) \nabla f(x_i)^T$$

3. Compute the spectral decomposition $\hat{C} = W \Lambda W^T$. The first $d$ eigenvectors of $\hat{W}$ form the coordinate transformation matrix $\hat{W}_1$.

here $y$ is the coordinates of active subspaces while $z$ represents the inactive directions that function $f$ does not vary too much. Then we define the domain of active subspace as follows.

*Definition 4:* Suppose $\mathscr{X}$ denotes the domain of original parameter space. For $\forall x \in \mathscr{X}$, $x = W_1^T y + W_2^T z$. Then the domain $\mathscr{Y}$ of *active(reduced) subspace* is defined as

$$\mathscr{Y} = \{ y : y = W_1^T x, x \in \mathscr{X} \} \tag{9}$$

We could construct the interpolation $g$ in reduced subspace to approximate the function $f$. The interpolation $g$ is built up by conditional expectation and will be talked in details in Section III-B

$$f(x) \approx g(W_1^T x) \tag{10}$$

### B. Dimension reduction of Δ-DOGS

In this section, we consider identifying the location in the feasible domain $B$ with the function value less than or equal to $f_0$. The objective function only varies primarily along at most $d$ directions in the feasible domain.

*Definition 5:* The scalar function $f(x)$ has at most $d$ dominant directions of the input ($d < n$). Suppose $x_D$ is the $d$-dimensional column vector that represents the value of input along $d$ dominant coordinates. For $\forall \gamma$ and $\forall x, y \in \mathbb{R}^n$ there exists $\delta$ such that if $\|x_D - y_D\| < \delta$, then $|f(x) - f(y)| \leq \gamma$.

The goal of dimension reduction scheme of Δ-DOGS algorithm is to identify the global minimum of the objective function which only has most variation in a few $d$ directions. This task is completed in three phases.

1. **First phase: Active Subspace.** In first phase, we apply the active subspace method to determine the $d$ directions that have the most variability of the objective function. The original data set $S$ would be mapped to the $d$-dimensional smaller set active subspace $\mathscr{Y}$. Each coordinate of the active subspace $\mathscr{Y}$ is a linear combination of original parameters in $\mathscr{X}$.
2. **Second Phase: Δ-DOGS.** In second phase, the regular Δ-DOGS is performed on the active subspace $\mathscr{Y}$ to determine the potentially minimizer $y_r \in \mathbb{R}^d$. This step aims at providing a priori knowledge about which subregion of the parameter space $\mathscr{X}$ might have global minimum. Notice that this step avoids the huge amount of memory storage to build the Delaunay triangulation in high dimensional space as needed for Δ-DOGS.
3. **Third Phase: DR-Δ-DOGS.** To make use of this minimizer, we define a inverse mapping to transform the point from $d$-dimensional active subspace $\mathscr{Y}$ to $n$-dimensional full-model $\mathscr{X}$. This inverse mapping requires another response surface constructed with data points in $\mathscr{X}$.

For second phase, we first construct a new interpolation that is needed for Δ-DOGS optimization in active subspace $\mathscr{Y}$. We implement Natural Polyharmonic Spline interpolant strategy which has been widely used to interpolate scattered data by minimizing the measure of curvature of the interpolant. The approximate value $\hat{f}(x_r)$ at the image of evaluated points after mapping is calculated based on conditional expectation of the interpolant value.

*Definition 6:* The value of the interpolation in reduced subspace is defined as

$$\begin{gathered} P_r(y) = \hat{f}(y) = \frac{1}{m} \sum_{i=1}^{m} f(x_{e_i}) \\ \text{subject to } W_1^T x_{e_i} = y, \quad y \in \mathscr{Y}, \ x_{e_i} \in \mathscr{X} \end{gathered} \tag{11}$$

The approximate function value $\hat{f}(y)$ is the expectation of the inverse image $x \in S$ in original parameter space $\mathscr{X}$.

Notice that the reduced model $\mathscr{Y}$ is also a manifold, thus another mesh grid scheme is applied in the reduced subspace. Each time the mesh grid $L$ in original parameter space $\mathscr{X}$ get refined, the mesh grid $\ell$ in reduced subspace $\mathscr{Y}$ will also be refined to accelerate the convergence of Δ-DOGS to the global minimum in reduced model $\mathscr{Y}$.

*Definition 7:* For the current work we consider one-dimensional active subspace. Since active subspace is linear mapping, the range of parameter $y \in \mathscr{Y}$ is simple to determine in one-dimensional space, denoted as $a'$ and $b'$. The Cartesian mesh grid in active subspace $\mathscr{Y}$, denoted as $B_\ell$, is defined as

$$B_\ell = \left\{ x | x_\iota = a'_\iota + \frac{1}{N}(b'_\iota - a'_\iota) \cdot z_\iota, \quad z_\iota \in \{0, 1, \ldots, 2^\ell\} \right\}$$

By implementing Δ-DOGS optimization, we obtain the minimizer $y_r$ of continuous function $s_c(x)$ in reduced subspace, which works as a priori knowledge to indicate which subregion of the parameter space potentially has the candidate minimizer.

Since the function evaluations are performed in original parameter space $\mathscr{X}$, it is needed to approximate $y_r$ in original parameter space $\mathscr{X}$. The normal way to overcome this task is by calculating $W_1 W_1^T y_r$. However, $W_1 W_1^T y_r$ is actually a point that lies on $d$-dimensional manifold in the original parameter space $\mathscr{X}$. It is obvious that this approach could possibly lost some amount of function behavior. Thus we propose an inverse mapping that retransform $y_r$ to original parameter space $\mathscr{X}$ based on the goal of minimizing

the surrogate of objective function in $\mathscr{X}$ [20]. This surrogate has the property to This inverse mapping is constructed by solving a inequality constrained minimization described below.

In inequality constrained optimization, the objective function is defined as discrete search function $s_d(x)$ which has a similar structure to continuous search function. $s_d(x)$ is constructed by the interpolant function $P(x)$ in $\mathscr{X}$ and a new uncertainty function $u(x)$, the distance-uncertainty function. The uncertainty function $u(x)$ is the distance of $x$ to its nearest neighbor in the evaluated points set $S$ defined as follow.

*Definition 8:* Suppose $S = \{x_1, x_2, ..., x_N\}$ denotes the evaluated points set in original parameter space. For $\forall x \in \mathscr{X}$ the uncertainty function $u(x)$ is defined as

$$u(x) = \text{dist}(x, S) = \min_{z \in S} \|x - z\| \tag{12}$$

Then the search function $s_d(x)$ is defined as

$$s_d(x) = \frac{P(x) - f_0}{\text{dist}(x, S)} = \frac{P(x) - f_0}{\min_{z \in S} \|x - z\|} \tag{13}$$

It is obvious that the distance-uncertainty function $u(x)$ is continuous and differentiable inside the Voronoi cell of every evaluated point $x \in S$. The key properties of $u(x)$ are: 1) $u(x) \geq 0 \ \forall x \in \mathscr{X}$, and $u(x_i) = 0 \ \forall x_i \in S, \ i = \{1, \ldots, N\}$; 2) Since the point-wise distance in $\mathscr{X}$ is bounded, and $\max u(x)$ is achieved on the boundaries of box domain $B$, thus $u(x)$ is Lipschitz continuous with Lipschitz constant $L_u$.

$$\|u(x) - u(x')\| \leq L_u \|x - x'\|, \ \forall x, x' \in \mathscr{X} \tag{14}$$

*Definition 9:* Determine the minimizer of Δ-DOGS $y_r$, and establish the discrete search function as stated in Definition 8. Given a slack tolerance variable $\varepsilon$, the inequality constrained minimization is defined as follow

$$\begin{aligned} &\min \ s_d(x) = \frac{P(x) - f_0}{\text{dist}(x, S)} \\ &\text{with } \|W_1^T x - x_r\| \leq \varepsilon \end{aligned} \tag{15}$$

This inequality constrained optimization is solved by Sequential Least-Square Quadratic programming. The initial guess is defined by $x_0 = W_1 W_1^T y_r$. The slack variable $\varepsilon$ is a user-defined variable that how much amount of variation that we could tolerate. The larger $\varepsilon$ indicates that we allow searching more globally in $\mathscr{X}$. A detailed convergence analysis would be performed in the following Section IV.

Up till now we have presented all the essential elements of new algorithm. The framework of new algorithm is showed in Algorithm 3.

**Algorithm 3** Dimension Reduction of Δ-DOGS

0. Initialize $k = 0$, $L$, $\ell$ and the initial set of datapoints $S^0$, and calculate $f(x_i)$ for all $x_i \in S_0$.
1. Calculate or update the interpolating function $p^k(x)$ for all the points in $S^k$.
2. By Algorithm 2, calculate or update the uncentered covariance-like matrix $C$ and the coordinate transformation matrix $W_1^k$.
3. By Definition 6, establish the interpolating function $P_r^k(x)$ in reduced model, minimize the continuous search function (3) to obtain $y_r^k$ as a minimizer in reduced model.
4. Solve the inequality constrained minimization (15) to obtain $x^k$ as a minimizer of the response surface.
5. Determine $z^k$ as the quantization of $x^k$ on $B_{L_k}$. If $z_k \notin S^k$, $S^{k+1} = S^k \cup z_k$; otherwise, refine the mesh by increasing $L_k = L_k + 1$ and $\ell_k = \ell_k + 1$. Increase $k = k + 1$.
6. Repeat steps 1-5 until a point $x$ is found with $f(x) \leq f_0$.

## IV. Convergence Analysis

In this section, we analyze the convergence properties of Algorithm 3. Under the appropriate assumptions, we will establish the following property:

*Target achievability:* If the target is achievable, the algorithm will either: (a) find the feasible point with objective function equal or less than the target $f_0$ in a finite number of iterations, or (b) generate an infinite sequence of points that contain a point with function value equal to $f_0$.

First we establish the following theorem based on Definition 5: As long as the perturbation of the $d$ dominant coordinates are sufficiently small, the change of values in active subspace interpolant is also small enough.

*Theorem 1:* Suppose the perturbation of the dominant directions is small, $\|x_D - x'_D\| \leq \delta$. Let $x = W_1 y + W_2 z$ and $x' = W_1 y' + W_2 z'$. Then the difference of the interpolant in active subspace is also small and it is Lipschitz with constant $L_{P_r}$.

$$|P_r(y) - P_r(y')| \leq L_{P_r} \|y - y'\| \tag{16}$$

where $L_{P_r} = \frac{2C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \gamma}{\delta_0}$.

*Proof*: From assumption 5, the change of objective only varies on $d$ main directions, recall the spectral decomposition on matrix $C$ in Algorithm 2, the last $n-d$ eigenvalues are sufficiently small. Suppose there exists a sufficiently small value $\varepsilon_0$ such that

$$\sum_{i=n-d}^{n} \lambda_i \leq \varepsilon_0 \tag{17}$$

Suppose $W_1$ is the first $d$ orthonormal eigenvectors of matrix $C$. By equation (11), we have

$$P_r(W_1^T x) = \hat{f}(W_1^T x) = \mathbb{E}[f(x))] = \mathbb{E}[f(W_1^T y + W_2^T z] \tag{18}$$

As sufficiently many data points are collected, the interpolant $P_r(y)$ is a realization of response surface that approximate the objective function $f$. From Theorem 4.4 in [19] and equation (17), the difference of objective function and reduced interpolant is bounded by

$$\begin{aligned} |f(x) - P_r(W_1^T x)| &\leq C_1(1 + N^{-\frac{1}{2}})(\lambda_{n+1} + \ldots + \lambda_m)^{\frac{1}{2}} \\ &\leq C_1(1 + N^{-\frac{1}{2}})\varepsilon_0 \end{aligned} \tag{19}$$

Here $m$ is the number of data points in full-model to approximate the reduced interpolant value at $W_1^T x$.

From the assumption 5, for $\forall x, x' \in \mathscr{X}$ such that $\|x_D - x'_D\| \leq \delta$, we derive that

$$\begin{aligned} &|P_r(y) - P_r(y')| = |P_r(W_1^T x) - P_r(W_1^T x')| \\ &= |[P_r(W_1^T x) - f(x) + f(x)] - [P_r(W_1^T x') - f(x') + f(x')]| \\ &\leq |P_r(W_1^T x) - f(x)| + |P_r(W_1^T x') - f(x')| + |f(x) - f(x')| \\ &\leq 2C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \gamma \end{aligned} \tag{20}$$

As the variation of objective function along $n-d$ directions goes to zero, the value of $\varepsilon_0$ also goes to zeros. Therefore, with a small perturbation in the $d$ dominant coordinates of the input $x$, the difference of the interpolant $P_r(x)$ in active subspace also has a relative small change.

Suppose the difference between $\|y - y'\| = \delta_0$, then the equation (20) can be rearranged as

$$\begin{aligned} |P_r(y) - P_r(y')| &\leq \frac{2C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \gamma}{\delta_0}\|y - y'\| \\ &\leq L_{P_r}\|y - y'\| \end{aligned} \tag{21}$$

Hence we showed that the interpolant $P_r(y)$ in active subspace is also Lipschitz continuous. While the uncertainty function $u(x)$ also keeps the properties such as continuous and twice-differentiable as needed to prove the target achievability in [11]. It is established in [13] that Algorithm 1 is capable to converge to the point with the target value $f_0$. The results is shown in the Theorem 2.

*Theorem 2:* Suppose the Definition 5 holds and constructs the Lipschitz continuous reduced interpolant $P_r(x)$ as stated in Definition 6. Algorithm 3 will converge to the global minimum of the feasible domain $\mathscr{Y}$.

In active subspace $\mathscr{Y}$, Δ-DOGS could approach the target value $f_0$, i.e. for sufficiently many iterations $k$, we have

$$|P_r(x_r^k) - f_0| < \varepsilon_2 \tag{22}$$

Finally we prove that the Algorithm 3 is also target achievable in original parameter space $\mathscr{X}$ under appropriate assumptions.

*Theorem 3:* Suppose the objective function has most variability along $d$ directions and assumption 5 holds. Suppose a target value $f_0$ and box domain $B$ are given. The Algorithm 3 is target achievable if the reduced interpolant $P_r(x)$ is Lipschitz continuous.

*Proof*: Suppose the Algorithm 3 has been run sufficiently many iterations such that the target is achieved at iteration $k$ in active subspace. Suppose at iteration $k$, $x_r^k$ is the minimizer of Δ-DOGS scheme and $x^k$ is the minimizer of Algorithm 3. The function value $f(x^k)$ is evaluated at iteration $k$.

Then according to equation (22), we have

$$|P_r(x_r^k) - f_0| \leq \varepsilon_2 \tag{23}$$

Since $x^k$ is the minimizer of the inequality constrained minimization, $x^k$ satisfies the inequality constraint $\|W_1^T x^k - x_r^k\| \leq \varepsilon$. By Theorem 1, the active subspace interpolant is Lipschitz constant with $L_{P_r}$, we have

$$|P_r(W_1^T x^k) - P_r(x_r^k)| \leq L_{P_r}\|W_1^T x^k - x_r^k\| \tag{24}$$

From equation (19), we derive that

$$P_r(W_1^T x^k) \leq f(x^k) + C_1(1+N^{-\frac{1}{2}})\varepsilon_0 \tag{25}$$

From equation (23), we have

$$P_r(x_r^k) \geq f_0 - \varepsilon_2 \tag{26}$$

Combining equation (24), (25) and (26), and the target is achievable by Δ-DOGS,

$$\begin{aligned} L_{P_r}\|W_1^T x^k - x_r^k\| &\geq |P_r(W_1^T x^k) - P_r(x_r^k)| \\ &\geq |(f(x^k) + C_1(1+N^{-\frac{1}{2}})\varepsilon_0) - (f_0 - \varepsilon_2)| \\ &\geq |f(x^k) - f_0| + |C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \varepsilon_2| \end{aligned} \tag{27}$$

Rearrange the above equation then we derive that

$$\begin{aligned} |f(x^k) - f_0| &\leq L_{P_r}\|W_1^T x^k - x_r^k\| - |C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \varepsilon_2| \\ &= L_{P_r}\varepsilon + |C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \varepsilon_2| \end{aligned} \tag{28}$$

Thus as long as we pick sufficiently small value of $\varepsilon_0$, $\varepsilon$ and $\varepsilon_2$, the difference between $f(x^k)$ and $f_0$ would be sufficiently small. The target value is achievable for Algorithm 3 with the above appropriate assumptions.

## V. Results

In this section, the new dimension reduction algorithm has been applied to the following synthetic function to test the performance. The first test function is nonconvex and the second test function is convex. For the following test problems, the initial number of subintervals for each coordinate is set to be 8. The Algorithm 3 continues until 4 times of mesh refinement are performed. The discretization error would be less than 0.002 when the algorithm terminates, i.e. $\|x^k - x^*\| \leq 0.005$. We initialize $\varepsilon = 0.2$ and iteratively reduce it to zero. For each iteration we reduce it by 0.001 until $\varepsilon = 0$.

The performance of the DR-Δ-DOGS algorithm is measured by the number of function evaluations and the relative error defined as follows. Suppose the best minimum point obtained until iteration $k$ is defined as the candidate point at iteration $k$. Let $f_{min}$ denotes the best minimum obtained by optimization and $f_0$ denotes the global optimum value. The relative error is defined as

$$\varepsilon = \frac{f_{min} - f_0}{|f_0|},\ y_0 \neq 0$$

The initial data points in $S^0$ are constructed with $3n+3$ points that are uniformly drawn from the parameter space $\mathscr{X}$.

The number of dimension of active subspace is set to be one. And the active subspace is obtained by uniformly random sampling the gradient of the objective function.

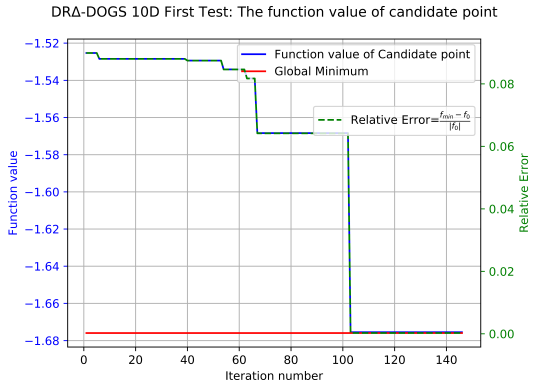


Fig. 2: $f_1(x)$ on Candidate point.

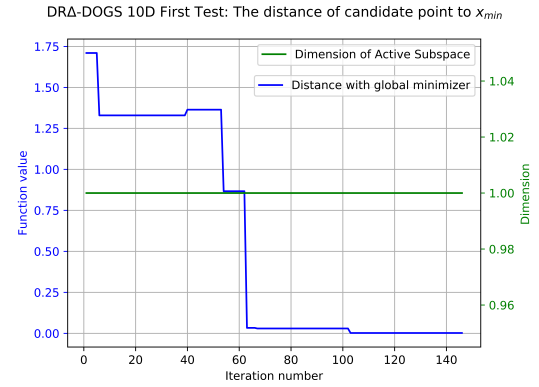


Fig. 3: Distance to global minimum.

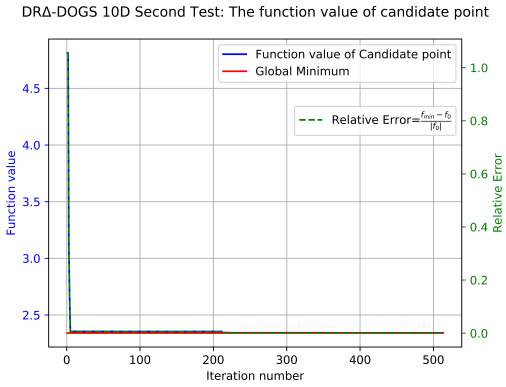


Fig. 4: $f_2(x)$ on Candidate point.

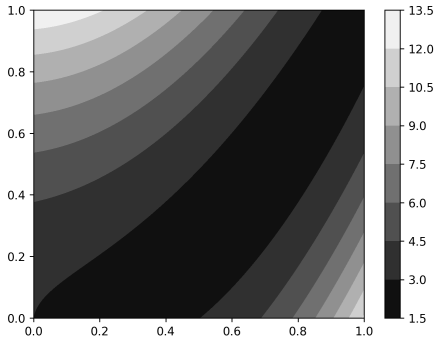

Fig. 5: Contour plot of the most variability of $f_2(x)$ on the manifold constructed by coordinates $x_1$ and $x_2$.

To approximate the first $d$ eigenvectors of $C$, [19] recommended to have $M$ samples of gradient sampling. Here $M = \alpha d \log(n)$ and $\alpha$ is an oversampling factor that chosen as 10, the other parameters are set as $d = 1$ and $n = 10$. As we have mesh grid refined, we would increase the gradient samples as there are more grid points in the parameter space. Thus the number of gradient sampling is defined as

$$M = \alpha d \log(n) L_k \tag{29}$$

The first test function is constructed by Schwefel function and quadratic function. The first component of (30) is constructed by Schwefel function and the rest coordinates are quadratic that have small contribution to the objective function. The first test function is defined as follows.

$$f_1(x) = -\frac{x_1}{2}\sin(500|x_1|) + \sum_{i=2}^{10}(0.001)\cdot i \cdot x_i^2 \tag{30}$$

It has several properties: (a) It has most variability along the coordinate $x_1$; (b) It is continuous, nonconvex and twice-differentiable; (c) It is defined on a box domain $x \in [0,1]^{10}$. (d) The minimizer $x^* = [0.8419, 0, ..., 0]$ and the target value $f_0 = -1.675936$. Apply Algorithm 3 on the first test function with $n = 10$. The target value $f_0 = -1.6759$ is achieved by 103 iterations with relative error 0.025%. The distance of candidate point to the global minimizer also converges to zero as shown in Fig. 2 and Fig. 3.

The second test function is a combination of exponential function, Rosenbrock function and quadratic function. The first component of (31) is the exponential function of $x_1$ and $x_2$. The second component is similar to Rosenbrock function but we decrease the variation of quadratic term by reducing the coefficient from 100 to 10. The third component is quadratic function of the rest 8 coordinates that have small contribution to the objective function.

$$f_2(x) = e^{0.2x_1} + e^{0.2x_2} + 10(x_2 - x_1^2)^2 + (x_1 - 1)^2 + 0.001 \cdot \sum_{i=3}^{10}(x_i - 0.1\cdot i)^2 \tag{31}$$

The second test function has several properties:(a) The most variability direction is along the combination of two coordinates $x_1$ and $x_2$;(b) It is continuous, convex and differentiable; (c) The minimizer is $x^* = [0.512, 0.723, 0.3, 0.4, ..., 1]$ and the target value $f_0 = 2.34128$. The contribution of the first two components are illustrated in Fig 5. The target value $f_0$ is achieved by 413 iterations with relative error 0.0532% in Fig. 4

Table I shows the results of applying new DR-Δ-DOGS to the above two test problems.

| Test function | No. param. | stopping criterion | final error ($\varepsilon$) | No. Eval. for 1% error |
|---|---|---|---|---|
| (30) | 10 | 146 | 0.025% | 52 |
| (31) | 10 | 465 | 0.0532% | 246 |

TABLE I: Experiment results of Algorithm 3.

## VI. Conclusions

This paper introduces a modification to the Delaunay-based derivative-free optimization algorithm scheme Δ-DOGS as proposed in [13], [14], identifying the active subspace and perform dimensionality reduction on the active subspace. The new scheme, Algorithm 3, has three main modifications as compared with the original Δ-DOGS algorithm:

- In this paper we have extended Delaunay-based derivative-free algorithm Δ-DOGS to high dimensional problem. Previously Δ-DOGS is restricted by the number of design parameters because of the unaffordable computational cost to construct the Delaunay triangulation. Under appropriate assumptions on the objective function, the new Algorithm 3 is convergence provable to identify the point on which the objective function achieves the target value.
- We proposed a new inverse mapping method that minimizes the global surrogate subject to inequality constraint. This optimization links these two subspaces while enforcing the points $x$ in full-model projected to the minimizer of the response surface in reduced space $y_r$.
- The objective function could have different variance along different coordinates. However, Δ-DOGS scales poorly with the dimension of the objective function because it treats each coordinate has the same importance in their contribution to the function. This new algorithm could mitigates this effect by projecting the original parameter space $\mathscr{X}$ to the active subspaces on which the coordinates all have approximately equal variance.

In future work, this framework will be applied to the application-based problems with objective function with its most variability along more than one directions. Increasing the dimension of the active subspace could make current algorithm handle more complicated objective functions. The vertices of active subspace $\mathscr{Y}$ could be found by algorithm provided in [21] subject to linear mapping. And the new scheme will be applied to hydrofoil shape optimization problem [3] with seven parameters. In our previous work we presented a hybrid optimization scheme that combined Delaunay-based derivative-free algorithm and a derivative-based local optimization scheme to accelerate local convergence in parameter space [22]. The presented dimension reduction technique for Delaunay-based optimization scheme will be incorporated into Δ-DOGS family of schemes [12], [11], [13] to enable them to handle problems with more than ten variables.

## Acknowledgement

The authors would like to thank Pooriya Beyhaghi for providing critical insights for this work. The authors also gratefully acknowledge funding from AFOSR FA 9550-12-1-0046 in support of this work and Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and

## References

[1] Torn, A., & Zilinskas, A. (1989). Global optimization. Springer-Verlag New York, Inc..

[2] Stephens, C. P., & Baritompa, W. (1998). Global optimization requires global information. Journal of Optimization Theory and Applications, 96(3), 575-588.

[3] Meneghello, G., Beyhaghi, P., and Bewley, T. : Simulation-based optimization of the hydrofoil of a flying catamaran. Journal of Ocean Engineering and Marine Energy. Under review.

[4] Saltelli, Andrea, Karen Chan, and E. Marian Scott, eds. Sensitivity analysis. Vol. 1. New York: Wiley, 2000.

[5] Adjengue, L., Audet, C., & Yahia, I. B. (2014). A variance-based method to rank input variables of the mesh adaptive direct search algorithm. Optimization Letters, 8(5), 1599-1610.

[6] Wold, Svante, Kim Esbensen, and Paul Geladi. Principal component analysis. Chemometrics and intelligent laboratory systems 2.1-3 (1987): 37-52.

[7] Roweis, Sam T., and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. science 290.5500 (2000): 2323-2326.

[8] Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492.

[9] Wild, S. M., Regis, R. G., & Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions. SIAM Journal on Scientific Computing, 30(6), 3197-3219.

[10] Booker, A. J., Dennis Jr, J. E., Frank, P. D., Serafini, D. B., Torczon, V., & Trosset, M. W. (1999). A rigorous framework for optimization of expensive functions by surrogates. Structural optimization, 17(1), 1-13.

[11] Beyhaghi, P., Cavaglieri, D., & Bewley, T. (2015). Delaunay-based derivative-free optimization via global surrogates, part I: linear constraints. Journal of Global Optimization.

[12] Alimo, S. R., Beyhaghi, P., & Bewley. T. : Delaunay-based Derivative-free Optimization via Global Surrogates, Part III: nonconvex constraints. Journal of Global Optimization. Under review.

[13] Beyhaghi. P., & Bewley. T.: Implementation of Cartesian grids to accelerate Delaunay-based Optimization. Journal of Global Optimization. Under review.

[14] Alimo, S. R., Beyhaghi, P., & Bewley, T. : Implementation of dense lattices to accelerate Delaunay-based optimization. Optimization Methods and Software Journal, Under review.

[15] Alimo, S., Beyhaghi, P., Meneghello, G., & Bewley, T. (2017): Delaunay-based optimization in CFD leveraging multivariate adaptive polyharmonic splines (MAPS). In 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (p. 0129).

[16] Wahba, Grace. Spline models for observational data. Society for industrial and applied mathematics, 1990.

[17] Gramacy, R. B., & Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. Journal of the American Statistical Association, 103(483), 1119-1130.

[18] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Advances in neural information processing systems (pp. 2951-2959).

[19] Constantine, P. G., Dow, E., & Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4), A1500-A1524.

[20] Lukaczyk, T. W., Constantine, P., Palacios, F., & Alonso, J. J. (2014). Active subspaces for shape optimization. In 10th AIAA Multidisciplinary Design Optimization Conference (p. 1171).

[21] Mattheiss, T. H., & Schmidt, B. K. (1980). Computational results on an algorithm for finding all vertices of a polytope. Mathematical Programming, 18(1), 308-329.

[22] Alimo, S. R., Beyhaghi, P., & Bewley, T. R. (2017, December). Optimization combining derivative-free global exploration with derivative-based local refinement. In Decision and Control (CDC), 2017 IEEE 56th Annual Conference on (pp. 2531-2538). IEEE.